10/538369
PCT/IB2003/005625

Rec'd PCT/PTO JUN 2005

MODULAR INTEGRATION OF AN ARRAY PROCESSOR WITHIN A SYSTEM ON CHIP

The present invention relates to processing systems on an integrated circuit that include an array processor as a functional unit or coprocessor, and particularly to integrated systems that include a reconfigurable array processor.

An embedded system is some combination of hardware or software that is specifically designed for a particular purpose or application within an overall system, and may be fixed in capability or programmable. A mobile phone may, for example, have a power saving integrated circuit (IC) or "chip" operable only with its respective type of phone and devoted exclusively to controlling the display and other elements to conserve power.

The same mobile phone typically includes a digital signal processing integrated circuit, which executes the functions on a digital portion of the radio. In order to adapt to different and/or changing radio broadcast formats of an incoming signal, programmable radios would be desirable. However, digital radio processing functions can entail high data sample rates, along with high computational loads, that are typically impractical to implement on programmable hardware.

A typical approach to accommodate the computational load within the capabilities of the

programmable hardware is to design hardware acceleration modules that specialize in efficient computation of high-data rate and/or computational rate algorithms. The accelerators may be interfaced with the programmable processor using a number of techniques, each of which allow the programmable processor to control the operation of the accelerator, as well as to properly schedule the data to be exchanged with the accelerator. For instance, a general purpose DSP or other host may have a set of internal register addresses that are visible within the instruction set of the processor, but are mapped to input and output ports of a coprocessor interface. The accelerator inputs and outputs may be connected to this interface, and process data under control of the programmable processor. In this way proper data exchange is programmable by the general purpose device.

In another approach the general purpose programmable host or DSP allows new, high-speed functional units to be inserted into its datapath. The functional unit responds to instruction operation codes provided by the hierarchical controller, and exchanges data with internal register files and other units according the datapath configuration specified by the hierarchical controller.

While these approaches succeed in offloading excess computational loads from a programmable processor, they rely on accelerators with limited or no programmability to execute the computation-intensive tasks. In this manner an important element of the programmability has been lost.

The present invention is directed to the integration of an array processor as a reconfigurable accelerator to a host or main processor, the array processor greatly exceeding the execution processing capacity of the host processor. The coprocessor includes a two-dimensional array of processing cells. The coprocessor is communicatively connected to the host processor by an interface module that has a mechanism for reconfiguring information paths between itself and respective cells on a periphery of the array.

In another aspect, this invention relates to a host or main processor's functional unit, where the host processor is preferably a very long instruction word (VLIW) processor, and the functional unit preferably embodies a two-dimensional array of processing cells having an interface by which information paths to the array through respective cells on a periphery of the array can be reconfigured.

Details of the invention disclosed herein shall be described below, with the aid of the figures listed below, in which same or similar components are denoted by the same reference numbers over the several views:

FIG. 1 is a block diagram illustrating a processor/co-processor arrangement in accordance with the present invention

FIG. 2 is a schematic diagram showing an example of a device having an embedded array processor in accordance with the present invention;

FIG. 3 is a block diagram of an implementation of the array processor of FIG. 2 as a functional unit within a VLIW processor; and

FIG. 4 is a set of flow diagrams that depict exemplary flow of processing in initializing and updating of programs to be executed on the array processor of FIG. 3

FIG. 1 depicts an example of a connection arrangement 10 between a general-purpose digital signal processor (DSP) or micro-controller 20 and its closely-coupled co-processor 30, implemented as a two-dimensional array. The co-processor 30 assists the DSP 20 in performing certain types of operations. The execution speed of the co-processor 30, often expressed in millions of instructions per second (MIPS), is faster than that of the DSP 20. Accordingly, in partitioning functionality between the processors, the co-processor would embody the high-MIPS signal chain. The co-processor 30 is communicatively connected to the DSP 20 by and interface module 40. The DSP 20 utilizes a memory system 50. In one embodiment, the DSP 20 and its co-processor 30 communicate directly by means of the interface module 40. In another embodiment, the interface module 40 is communicatively connected to the memory system 50 to thereby provide a communications path, or and additional communications path, between the DSP 20 and the co-processor 30. In the latter embodiment, processor synchronization is implemented in preferably one or more of the modules 20, 30, 50.

FIG. 2 shows an exemplary embodiment of an apparatus that may be configured to incorporate the

arrangement 10 shown in FIG. 1. A receiver 100, such as one in a broadcast or cable television receiver, local area network wireless receiver or mobile phone receiver, contains an IC 102. The IC 102 includes an embedded array processor 106. An array processor is a processor capable of executing instructions that operate on input that may consist of arrays. The embedded array processor 106 has a two-dimensional rectangular array 108 and a mechanism or interface 110 which is shown in FIG. 2 to surround the array 108 on all four edges. The two-dimensional array 108 is composed of processing cells 112.

The IC 102 may, for example, be configured in accordance with the arrangement 10 in FIG. 1, where the array 108 is implemented as the array 30 and the interface 110 corresponds to the interface module 40. As will be discussed below, other additional alternatives for implementing IC 102 are contemplated.

Preferably, inter-cell connection within the array 108 is such that each cell 112 is connected only to cells 112 whose column is the same and whose row is immediately adjacent, and only to cells 112 whose row is the same and whose column is immediately adjacent, to realize a "nearest neighbor" connection architecture, as shown in FIG. 2 of commonly owned U.S. Patent Publication No. 2003/0065904, filed October 1, 2001, (hereinafter the '904 application), the entire disclosure of which is incorporated herein by reference. Since inter-cell connection is purely nearest-neighbor, the array offers the flexibility of being scalable.

In one embodiment, the interface 110 has border cells 114 connected to each respective processing cell 112 on the periphery of the array 108, each border cell 114 having a buffer 116. The periphery preferably consists of those processing cells 112 which are located on the array edges, i.e., in at least one of the first row, last row, first column and last column. Since internal array connection cell-to-cell, under the nearest neighbor scheme, leaves two neighbors missing for each corner cell 112 and one neighbor missing for each other cell 112 on array edges, the missing connections are each made to a corresponding border cell 114.

Further included in the interface 110 are input/output (I/O) pads 118, one for each border cell 114, and a crossbar network 120 for reconfigurably connecting each I/O pad 118 one-to-one to a corresponding border cell 114. For each such connection an information path is formed. FIG. 2 shows an information path 122 that includes an I/O pad 118, the crossbar network 120 and a border cell 114. Reconfiguring a path causes the path to traverse either a different border cell 114, a different I/O pad 118, or both. The path 124 is a reconfiguration of the path 112 to traverse a different border cell 114. Reconfigurable routing can alternatively be accomplished via a local selection mechanism in each border cell, rather than by a crossbar network.

In a preferred embodiment, the array processor 106 is a systolic processing array, a special-purpose system which can be likened to an assembly line for input operands, although operations typically proceed not in a

strictly linear direction but in changing directions. In a two-dimensional array of processing cells, differing mathematical operations are performed on the data by different cells, while data proceeds in an orderly, lock-step progression from one cell to another. An example of a systolic array would be one that multiplies matrices. Entries of a row are multiplied by corresponding entries of a column, and the products are summed to produce an ordered column of sums. Efficiency is achieved by arranging operations to be performed in parallel, so that the results are produced in the fewest clock cycles. The '904 application provides another example of a systolic processing array, implementing a 32-tap real finite impulse response (FIR) filter. The filter is enhanced by concatenating other levels, two-dimensional and otherwise, to the original two-dimensional array, border cells being connected to processing cells on the periphery of each level. Such an enhanced array, connected by the border cells 114, is also within the intended scope of the present invention.

In one embodiment, the border cells 114 not only provide input to the array 108. They also provide results of array processing to the I/O pads 118. The border cells 114 receive these results by neighbor to neighbor conveyance from the processing cells 112 producing the results. Optionally, the border cell 114 may validate the results and output a data valid signal to the external process, such as the DSP 20.

In a preferred embodiment, the IC 102 includes a memory, such as in memory system 50, from which array

programs are downloaded by means of a bus 113 to corresponding processing cells 112. The memory is preferably a random access memory (RAM) or other writeable storage device so that updated array programs can be provided, as by an array generator external to the receiver 100.

The system controller which may be an external processor passes array programs to a master cell 126 of the embedded array processor 106 over a configuration bus such as the random access configuration bus shown in FIG. 16 of the '904 application. As discussed in the pending, commonly owned patent application entitled "DATAFLOW-SYNCHRONIZED EMBEDDED FIELD PROGRAMMABLE PROCESSOR ARRAY," based on Philips disclosure 703366, hereinafter the "EFPPA application," the entire disclosure of which is incorporated by reference herein, the master cell 126 forwards the array programs to the appropriate processing cells 112 at system initialization or upon reconfiguration, e.g. implementation of a new algorithm for the processing array 106. Due to the parallelism inherent in systolic processing, some of the processing cells 112 may receive identical programs. An identical program may, for example, be downloaded to a subset of the processing cells 112 such as subset 115 shown in FIG. 2. The EFPPA application further discusses processing by the border and master cells and a preferred implementation using a Kahn process network.

The array processor 106 performs mathematical operations whose timing is based on a flow of input

operands along the paths providing the operands to the array 108.

Array programs may be prepared using a graphical user interface (GUI) that can edit and show the code to be downloaded to RAM on the IC 102 and then to each programming cell 112.

In an alternative exemplary implementation 300 of the embedded array processor 106 of FIG. 2, FIG. 3 depicts a host VLIW processor 302 as a component of an EFPPA 304 of the "in circuit" programmable type. EFPPA 304 is implemented on an IC 306 contained within a receiver 308. The host VLIW processor 302 is connected to a chip development platform 309, and, in particular, to an array program generator 310 and a compiler 312 within the platform 309. The array program generator 310 is further connected to a graphical user interface 314 of the platform 309.

The VLIW processor 302 includes an instruction memory 316, and instruction issue register 318, a shared, multiported register file 320. Also included within the processor 302, and, connected to both the file 320 and the register 318 at corresponding issue slots, are a plurality of functional units. Details of this VLIW architecture are provided in commonly owned U.S. Patent No. 5,974,537, filed October 26, 1999, (hereinafter the '537 patent),the entire disclosure of which is incorporated herein by reference. The functional unit 322 can be realized, for example, as the embedded array processor 106 of FIG. 2 in the present application, with the IC 306 corresponding to IC 102, and with the receiver 308 corresponding to receiver 100. In

the '537 patent, the functional unit 322 executes floating point instructions, although the unit 322 is not confined to any particular type of processing. For example, a two-dimensional array is disclosed in the '904 application to perform finite impulse response (FIR) filtering and fast Fourier Transforms (FFT's) useful in channel decoding and other applications.

FIG. 4 demonstrates exemplary flow of processing in initializing and updating of programs to be executed on the array processor 322 of FIG. 3. At system initialization, array programs for each of the processing cells 112 generated by the array program generator 310 (step 402) are downloaded to a RAM 324 on IC 306 (step 404). A system controller (not shown) subsequently downloads the array programs to the master cell 126 which distributes them to the corresponding array cells 112. The master cell 126 accordingly transmits a plurality of array programs to corresponding predetermined subsets of the processing cells 112, the cells in each subset of one or more cells receiving an identical array program.

When an array program is updated, as by a user of the chip development platform 309 through interactive utilization of the GUI 314 and by means of the array program generator 310(steps 406, 408), changes in the program may affect the timing of functional unit 322 input and/or output. The compiler 312 needs to know this timing change for scheduling purposes in forming the VLIW instruction. The array program generator 310 therefore updates this I/O timing data and transmits it to the compiler 312 (step 410). The updated array program is

downloaded (step 412), as described above with regard to system initialization. The array program generator 310 determines whether the program change affects a steady state connection pattern of the interface 110. The steady state pattern defines, for example, which I/O pads 118 are connected to which border cells 114 at which stages of a mathematical operation, i.e., the mathematical operation may accept input operands at the array periphery at multiple stages of the operation. If the program update changes the steady state pattern (step 414), the array program generator 310 sends a reconfigure signal to the functional unit 322 (step 416). Preferably, the signal is received by the master cell 126, which then effects the needed connection timings in the crossbar switch 120.

Although array program functionality has been described in the context of the VLIW processor 302 of FIG. 3, the same functionality, except for the timing data protocol, applies as well to the coprocessor arrangement 10 of FIG. 1. In fact, even the timing data protocol applies if the co-processor is implemented as a VLIW processor.

While there have been shown and described what are considered to be preferred embodiments of the invention, it will, of course, be understood that various modifications and changes in form or detail could readily be made without departing from the spirit of the invention. For example, alternatively implemented, the system controller 104 and RAM may instead reside within the embedded array processor 106. It is therefore intended that the invention be not limited to the exact forms described and illustrated, but should be constructed to

cover all modifications that may fall within the scope of the appended claims.